GOLDEN RATIO SCHEDULING FOR LOW DELAY FLOW CONTROL IN COMPUTER NETWORKS

S. S. Panwar[1], T. K. Philips[2], and M. S. Chen[2]

[1]Department of Electrical Engineering and Computer Science, Polytechnic University, 333 Jay Street, Brooklyn NY 11201

[2]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

ABSTRACT

We present a scheme for flow control based on the Golden Ratio Policy of Itai and Rosberg [5] that requires very few buffers and guarantees low end to end delays. Messages are formed into equal length packets and then transmitted in accordance with a cyclic schedule. The scheme is very well suited to an Asynchronous Time Division Multiplexing (ATM) environment as it allows the networks capacity to be allotted to sessions in any desired proportion. The implementation is quite simple.

I. INTRODUCTION

Consider a network of computers (or nodes) interconnected by transmission links (or edges) communicating with each other via a store and forward mechanism. The virtual circuit that is set up between two communicating processes in distinct computers is called a *session*. Each link may carry traffic from hundreds of sessions, and these sessions must be given their fair share of the network's capacity. The task of controlling the entry and forwarding of messages in such a network is referred to as *flow control*.

The primary objective of a flow control policy is to ensure that the various sessions get a fair share of the networks capacity (under some appropriate measure of fairness). Implementational considerations lead us to conclude that the following properties are also desirable:

1. Low buffer requirements.
2. Low end to end delays.
3. The ability to operate in a decentralized manner.
4. The ability to take down and bring up sessions in a simple manner.
5. Stability in heavy traffic.

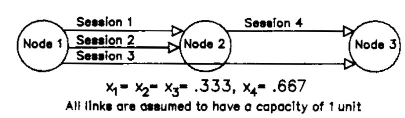

Figure 1. A 3 node network with 4 sessions.

In this paper we present a scheme to allocate available capacity to sessions in *any* desired proportion. A three node network with four sessions is shown in figure 1 along with the max-min fair rates [3] for each of the sessions.

Hahne [2] showed that if sessions were allowed to transmit in a round robin fashion and the window size (or the maximum number of transmitted messages for which acknowledgements have not been received) were sufficiently large, the max-min fair rates would automatically be enforced. This policy is completely distributed, and requires no inter processor communication. Unfortunately, the necessary window size (and consequently the number of buffers required for each session) is astronomically large. When the window size was restricted, max-min fairness was not always achieved, though the throughput of a session was lower bounded.

Another approach, adopted by Mukherji [7], is to have a transmission schedule that is cyclic. Each session is allotted at least one slot during each transmission cycle (or frame) and this lower bounds its throughput. This is well suited to applications such as digitized speech that require synchronous access to the network. A session may use more than one slot in every cycle, and uncommitted slots are allocated to sessions according to their instantaneous requirements. In addition, Mukherji provides upper bounds on the delays experienced by packets, and describes efficient algorithms for the construction of near optimal transmission schedules.

The policy we shall describe is cyclic too, and is designed so that transmission permits to the various sessions are fairly evenly spaced. This will be shown to lead to small buffer requirements, low end-to-end delays, and to allow sessions to be added and taken down simply. Session are allotted slots in accordance with their requirements, and consequently, this policy is well suited to an Asynchronous Time Division Multiplexing (ATM) environment.

The remainder of the paper is organized as follows. In Section 2, the Golden Ratio Policy is described. Section 3 is devoted to an exploration of its properties. The policy's performance is examined in section 4, and design considerations are the subject of section 5. Finally, in section 6, conclusions are drawn and some open problems are identified.

II. THE GOLDEN RATIO POLICY

Suppose we have S sessions on a given link labeled $1,2,\dots,S$, with capacity requirements $x_1 x_2,\dots,x_S$. Clearly $x_l \geq 0,\ 1 \leq l \leq S$, and we shall require $x_1 + \cdots + x_S = 1$. Normalizing the link capacity to 1 unit causes no loss of generality. If a link has some spare capacity, this can be allocated to a dummy session in whose slots nothing is transmitted. Fix $\varepsilon \geq 0$. (ε determines how accurately we wish the capacity requirements to be approximated.) Choose a frame (or cycle) length N sufficiently large so that $\max(x_l N - \lfloor x_l N \rfloor\ ,\ \lceil x_l N \rceil - x_l N) \leq N\varepsilon\ ,\ 1 \leq l \leq S$. Then define $X_l = \lfloor x_l N \rfloor$ or $\lceil x_l N \rceil$ subject to $X_1 + \cdots + X_S = N$. Let $\phi^{-1} \triangleq (\sqrt{5}\ -1)/2$. ϕ is also known as the Golden Ratio, and is related to the Fibonacci numbers via

$$F_k = \frac{\phi^k - (1-\phi)^k}{\sqrt{5}} .$$

Mark the points $\phi^{-1} \bmod 1,\ 2\phi^{-1} \bmod 1,\ \dots,\ N\phi^{-1} \bmod 1$ on a circle of circumference 1, dividing it into N intervals. Allot to session 1 the slots starting at $\phi^{-1} \bmod 1,\ 2\phi^{-1} \bmod 1,\ \dots,\ X_1\phi^{-1} \bmod 1$, to session 2 those starting at $(X_1 + 1)\phi^{-1} \bmod 1, \dots,\ (X_1 + X_2)\phi^{-1} \bmod 1$, and so on till all the slots have been allotted. Lastly, equalize the slot lengths. In effect, we have chosen a sufficiently long frame length, and then allotted slots within the frame in a reasonably regular manner.

Consider the following example: $\hat{S} = 4$, $x_1 = .3$, $x_2 = .3$, $x_3 = .2$, $x_4 = .2$, $\varepsilon = .1$. The network is assumed to have two nodes, say 1 and 2, and an edge joining them. All the sessions are assumed to start at node 1 and end at node 2. These requirements can be satisfied when $N = 5$, leading to $X_1 = 2,\ X_2 = X_3 = X_4 = 1$. The transmission sequence is 4 1 3 1 2 ... , as can be verified by direct computation.

We shall assume all the clocks in the network to run at an identical rate, implying the length of a transmission cycle to be the same at every node. They may, however, have fixed offsets relative to each other, making the transmission schedule at one node a time shifted version of that at another. Similarly, when sessions originating at distinct nodes share a link, the transmissions schedules of some sessions may be unavoidably offset from their transmission schedules on the preceding link even in the absence of clock skews. This necessitates the buffering of messages, as a message that arrives at a node may have to wait for some time before it can be forwarded.

Looking at the transmission sequence, the reason for the use of the Golden Ratio becomes clear. Not only are successive permissions to the sessions evenly spaced, but computing the transmission sequence is simple, enabling sessions to added or deleted without difficulty. The ability of the Golden Ratio policy to space successive permissions to transmit quite evenly has been explored in great detail [11, 12], and has found use in multiplicative hashing [6] and the design of TDMA protocols for the multiple access channel [5, 10].

III. PROPERTIES OF THE GOLDEN RATIO POLICY

With these preliminaries behind us, we can now state the main theorems. As the proofs are intricate, we shall refer the interested reader to [8]. Consider a session that passes through three of more nodes, such as session 3 in figure 1. It will be assigned certain slots in the transmission schedule at node 1 and these slots may or may not be synchronized with those allotted it at node 2. In general, they will not, and consequently we must have buffers to hold enqueued packets. The transmission schedule for the network pictured in figure 1 is shown in figure 2. Sessions 1, 2 and 3 are allocated two slots each at node 1 and sessions 3 and 4 share the slots at node 2 in the ratio 2:1. Note that a substantial skew exists between the two transmission schedules. The number of buffers required will depend on the skew between the two schedules, and is next computed in the special case when X_l is a Fibonacci number.

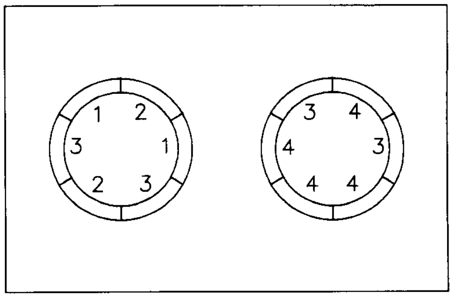

Figure 2. The Transmission Schedules at Nodes 1 and 2.

Theorem 1: Consider a Golden Ratio policy and suppose that

1. The total number of slots at every node is a Fibonacci number, say F_k,
2. The number of slots allotted to some session, say session 1, is a Fibonacci number, say F_{k_1}, $(1 \leq k_1 \leq k)$,
3. Session 1 is allotted the same number of slots at every node it encounters, and
4. One transmission buffer which can be shared by all the sessions is available.

Then if $k_1 = k$ or $k_1 \leq 2$, exactly 1 buffer is required by the session at every node on its path other than its source and destination, and if $k_1 < k$, exactly 2 buffers are required by it.

Theorem 2: Consider a Golden Ratio Policy and suppose that

1. The total number of slots at every node N, is <u>not</u> a Fibonacci number,
2. Schedules are globally synchronized (i.e. each slot starts and ends at exactly the same time at every node),
3. The number of slots allotted to some session, say session 1, is a Fibonacci number, say F_{k_1},
4. Session 1 is allotted the same number of slots at every node it encounters, and
5. One transmission buffer which can be shared by all the sessions is available.

Then if $k_1 \leq 2$, exactly 1 buffer is required by the session at every node on its path other than its source and destination, and in all other cases exactly 2 buffers are required by it.

When the total number of slots, N, is a Fibonacci number, but the number of slots allotted to a session , is <u>not</u> , the number of buffers required cannot be computed exactly. An upper bound however can be found, and is next presented.

Theorem 3: Consider a Golden Ratio policy and suppose that

1. The total number of slots at every node is a Fibonacci number, say F_k,
2. The number of slots, X_1, allotted to a session, say session 1, is not a Fibonacci number,
3. Session 1 is allotted the same number of slots at every node it encounters, and
4. One transmission buffer which can be shared by all the sessions is available.

Then the number of buffers required by the session at every node other than the source and destination is upper bounded by $2m$, where m is the smallest number of distinct Fibonacci numbers that X_1 can be represented as the sum or difference of; and $2m$ in turn is upper bounded by $2\left\lceil \frac{\lceil \log_\phi \sqrt{5}\, X_1 \rceil - 1}{2} \right\rceil$.

IV. PERFORMANCE ANALYSIS

If the conditions imposed by Theorems 1 and 2 are not satisfied, no simple method of computing the buffer requirement exists. Bounds on the number of buffers required may, however, be found either by Theorem 3 or via the staircase technique described in [8]. It is interesting to examine the bound on the number of buffers required for a fixed N as m is varied, as shown in the next two figures for $N = 89$ and $N = 150$. Note that 89 is a Fibonacci number while 150 is not. The curves are seen to be quite jagged, but tend to increase with m as can be seen from their envelope. Sharp dips are seen at Fibonacci numbers, even when the total number of slots is not a Fibonacci number. On looking at figure 3 we see that the staircase technique often, though not always, gives us the correct buffer requirements. For example, when $F_{k_1} = 55$, it bounds the buffer requirements by 3. Theorem 1, on the other hand tells us that only two buffers are required.

Let $B(X,N)$ denote the number of buffers required by a session that is assigned X out of N slots. Clearly, if $X = X_1 \pm X_2$, $B(X,N) \leq B(X_1, N) + B(X_2,N)$, and this bound is shown in dotted lines in figure 3. As N is then a Fibonacci number, we know the buffer requirements for all m that are Fibonacci numbers. The decomposition approach is seen to be useful mainly when m is large. When N is not a Fibonacci number, we do not know the buffer requirements except when $m = 1,2$ and N. This makes the decomposition approach useless for all but the largest values of m.

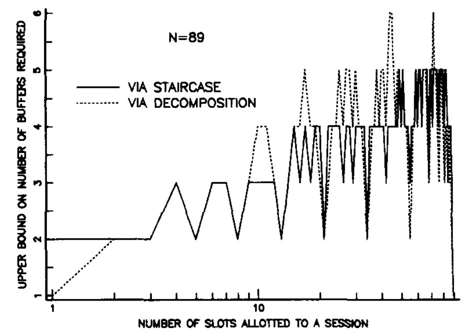

Figure 3. Bounds on the number of buffers required by a session

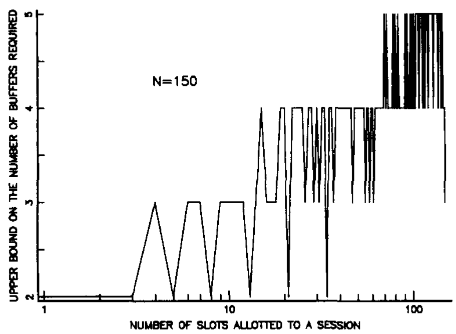

Figure 4. Bounds on the number of buffers required by a session

Lastly, we analyze network delays. There are two components of the delay- the first occurs at the source node, where incoming packets experience queueing delays, while the second occurs at intermediate nodes, where packets have to wait in buffers awaiting service. A detailed analysis of the first component can be found in [4]. Unfortunately, the expression for the waiting time presented there is complex, and is not very useful for quick "back of the envelope" calculations. If packets are assumed to arrive in a Poisson stream, a simple approximation for the mean waiting time exists [9]. Let N be the total number of slots, and let X_1 be the number of slots allotted to session 1. Let the distances between suceesive permits to the session be $d_1, \ldots, d_{X_1}$. Define the average interpermit distance $\bar{d} \triangleq N/X_1$, and let λ be the mean number of arrrivals in $\bar{d}$ slots. Then if $\lambda < 1$, the mean waiting time of a packet is approximately

$$\left\{ (1+\lambda)\sum_{i=1}^{X_1}\frac{d_i^2}{2N\bar{d}} + \frac{\lambda^2}{2[1-\lambda]} + \lambda\sum_{i=1}^{X_1}\frac{d_i(d_{i-1}-\bar{d})}{N\bar{d}} \right\} \frac{T}{X_1}$$

where T is the length of transmission cycle (in seconds).

The approximation was compared to a simulation. A million packets were processed, and the 99% confidence intervals were less than 1% of the mean. The error in the approximation is plotted in the next figure, for N=89, and is seen to be at most 2.5%.

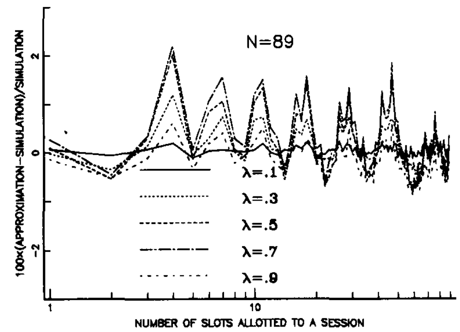

Figure 5. Error in mean waiting time.

The second component of the waiting time cannot be computed exactly, though it can be bounded and this is done in the next theorem

Theorem 4: Consider a Golden Ratio policy and suppose that

1. The total number of slots at every node is N, and F_k, is the largest Fibonacci number that does not exceed N,
2. The time taken for a transmission cycle is T seconds,
3. Sessions are allotted the same number of slots at every node they encounter,
4. Some session, say session 1 is a h hop session, is allotted X_1 slots. and requires b buffers at every node it passes through.
5. $F_{k_1}$is the largest Fibonacci number that does not exceed X_1.

Then the delay experienced by a packet sent by session 1, from the time it is transmitted from the initial node to the time it is received at the final node is upper bounded by $b(h-1)T \times F_{k-k_1+2}/F_k$

■

V. DESIGN CONSIDERATIONS

We now present two extensions of the Golden Ratio Policy that reduce the buffer requirements and eliminate the wastage of unused slots.

First let us consider the reduction of buffer requirements. From Theorem 3, the number of buffers needed for a session to fully utilize its slots grows at most logarithmically with the number of slots assigned to it. For example, when N = 1000, B(X, 1000) varies between 2 and 7 as X ranges from 2 to 999. This implies that the total number of buffers required at a node is larger when there are many sessions, each assigned a few slots, than when there are a few sessions, each of which is assigned a substantial fraction of the available slots. Therefore, ***multiplexing*** many sessions into larger sessions can reduce the buffer requirements.

To evaluate the effectiveness of multiplexing in buffer reduction, the differences between $B(X_1 + X_2 + \cdots + X_m, N)$ and $B(X_1, N) + B(X_2, N) + \cdots + B(X_m, N)$ are computed for every possible combination of $X_1 + X_2 + \cdots + X_m \leq N$, where m is the number of constituent sessions in a "multiplexed session." The results for three different m's (m = 2, 3 and 4) and N's (N = 89, 150, and 300) are summarized in two Tables 1 and 2. For example, about 40, 55, or 65 percent reduction can be obtained when m is 2, 3, or 4, respectively. Although results for large m are not available, multiplexing is clearly very effective in reducing buffer requirements.

N	m=2	m=3	m=4
89	39%	55%	64%
150	40%	71%	79%
300	42%	58%	67%

Table 1. Average buffer reduction from multiplexing

We now turn to the design aspect of multiplexing, and investigate the criteria used to decide the sessions that will benefit most from combining. This issue is addressed in [1]. A "Path-Based" scheme is proposed and compared with two other schemes: "Route-Based" and "Hop-Count" schemes. In the Path-Based scheme, two routes are combined together into a funnel-like entity when they enter a node if both their destinations and subsequent routes are identical. In the Route-Based scheme, sessions are combined together into a route when they have the same source, destination, and physical path. In the Hop-Count scheme, sessions are combined together if they have the same hop count.

The number of entities (or combined sessions) in the Path-Based scheme is shown to be better than in the Route-Based scheme by a factor of the average route length (measured in hops). The average route length typically lies between 3 and 4. The Hop-Count scheme, although has the least multiplexed entities, is not desirable because of possible congestion unfairness.

Secondly, we examine the possibility of reducing the wastage of unused slots. The Golden Ratio Policy is a schedule based scheme, and slots can be used only by the session assigned to them. This problem, which is typical of time division multiplexed (TDM) schemes can be solved by adopting statistical time division multiplexing (STDM). Two STDM schemes which mimic the Golden Ratio Policy are next presented. Note that some form of flow control is needed in any STDM scheme. A session may have different numbers of slots available to it on different links, depending on the number of idle sessions on each link, and its buffers can overflow if it has more slots on an upstream link than on a downstream link.

The first scheme, referred to as the "Dynamic Substitution" scheme, is based on a scheduling sequence and a replacement selection procedure. The scheduling sequence, say *SEQ*, is generated according to the Golden Ratio Policy. For example, $SEQ(i) = j$ implies that slot i is assigned to session j. At each slot, if the designated session is active, then one of its packets will be transmitted in the slot. If not, a replacement session will be selected to use the slot instead.

The selection of the replacement session can be carried out in many ways. For example, when session profiles are known, it may be desirable to select bursty sessions and avoid stream oriented sessions such as packetized voice, as stream oriented sessions do not usually need more bandwidth than they have been allotted. Among bursty sessions, moreover, the ones with heavier load should be given higher priority. Another intuitively appealing consideration, which tends to minimize the variance of the delay, is to select a session that currently has the longest queue. (This is the inverse of the classical "shortest queue routing" policy [13], which minimizes the mean waiting time of a customer). The second scheme, which we call the "Single Pointer" scheme, also uses the Golden Ratio Policy to generate a scheduling sequence, say SEQ. Unlike the Dynamic Substitution scheme, however, there is no fixed association between the indices in SEQ and the slot numbers. Instead, the actual transmission is completely governed by a single pointer in the following manner:

1. Move the pointer to the next entry of SEQ.
2. If the pointed session is active, transmit its packet.
3. If not, go to 1.

Lastly, we examine the conditions under which the Single Pointer Scheme and the Dynamic Substitution Scheme preserve the structure of the Golden Ratio Policy.

Lemma 1: Consider a Golden Ratio Policy with S sessions and let $\sum_{i=1}^{S} X_i \triangleq N$. Suppose that sessions are sequentially assigned, i.e., session 1 is assigned its slots first, session 2 is assigned its slots next, and so on till all the slots have been allotted. Then when session S or 1 is idle, the Single Pointer scheme is a Golden Ratio Policy with (S-1) sessions and cycle length $(N - X_S)$ or $(N - X_1)$.

Theorem 5: Consider a Golden Ratio Policy with S sessions, let $\sum_{i=1}^{S} X_i \triangleq N$, and suppose that sessions are sequentially assigned. Then if sessions i, i+1, ..., i+Z-1 are the only active sessions, the Single Pointer scheme is identical to a Golden Ratio Policy with Z sessions and a cycle length of $X_i + X_{i+1} \cdots + X_{i+Z-1}$.

Empirical results indicate that the Single Pointer scheme and the Golden Ratio Policy are identical more often than proven in Theorem 5. Moreover, when they are different, the differences are very small. Therefore, it is reasonable to extend the performance results derived in previous section to "approximate" the Single Pointer STDM scheme. In particular, numerical results indicate that when Y slots are allotted to idle sessions, the Dynamic Golden Ratio can reduce the mean waiting time occurs at the source node by a factor of (N-Y)/N.

VI. CONCLUSIONS

A schedule based flow control scheme constructed around the Golden Ratio policy that requires very few buffers and guarantees low end to end delays in a network has been presented. Various properties of the policy have been determined, and a number of implementational issues have been examined. Many open questions remain, however.

Tighter bounds on the buffer requirements when the number of slots assigned to a session is not a Fibonacci number would be of interest. An exact result would be of great interest to mathematicians, but is probably not determinable. Additionally, some understanding of the behavior of the interslot distances when the total number of slots is not a Fibonacci number would prove very useful.

Two other topics merit further investigation.

1. A provably optimal (under some suitable measure of optimality) scheme for combining sessions to reduce the buffer requirements.
2. An policy for assigning idle slots to sessions with enqueued packets so as to minimize the mean waiting time of a packet. We conjecture that the optimal policy will be to serve the session with the longest queue, but no proof of this has been found.

VII. REFERENCES

1 . Chen, M-S., B. Kadaba, and G. Grover, 'Efficient Hop By Hop Buffer Class Flow Control Schemes', Proc. GLOBECOM 87, Tokyo, Japan, 1987.

2 . Hahne, E.L., 'Round Robin Scheduling for Fair Flow Control in Data Communication Networks', Ph.D. Thesis, M.I.T, Cambridge, 1986.

3 . Gafni, E.M. and Bertsekas, D.P., 'Dynamic Control of Session Input Rates in Communication Networks', IEEE Trans. on Auto. Control, Vol. AC-29, No.11, 1984, pp 1009-1016.

4 . Hofri, A., and Rosberg, Z., 'Packet Delay under the Golden Ratio Weighted TDM Policy in a Multiple-Access Channel', IEEE Trans. on Info. Theory, Vol. IT-33, 1987, pp 341-349.

5 . Itai, A., and Rosberg, Z., 'A Golden Ratio Control Policy for a Multiple Access Channel', IEEE Trans. Auto. Control, Vol. AC-29, 1984, pp 712-718.

6 . Knuth, D.E., 'The Art of Computer Programming', Vol. 3, Addison Wesely, Reading, 1973, pp 506-544.

7 . Mukherji, U., 'A Schedule Based Approach for Flow Control in Data Communication Networks', Ph.D. Thesis, M.I.T, Cambridge, 1986.

8 . Panwar, S.S., Philips, T.K., and Chen, M.S., 'Golden Ratio Scheduling for Low Delay Flow Control in Computer Networks', IBM Research Report RC 13642, April 1988.

9 . Philips, T.K., and Panwar, S.S., 'Approximating the Mean Response Time Under the Golden Ratio Policy', submitted to IEEE Transactions on Information Theory.

10 . Rosberg, Z. and Sidi, M., 'TDM Policies in Multi-Station Packet Radio Networks', IBM Research Report RC 12781, May 1987.

11 . Sos, V.T., 'On the distribution mod 1 of the sequence $n\alpha$ ', Ann. Univ. Sci. Budapest Eotvos Sect. Math., Vol 1., 1958, pp 127-134.

12 . Swierczkowski, S., 'On successive settings of an arc on the circumference of a circle', Fund. Math., Vol. 46, 1958, pp 187-189.

13 . Winston, W., 'Optimality of the Shortest Line Discipline', J. Applied Probability, Vol. 14, 1977, pp181-189.